SIMULATING THE CONSTRUCTION PROCESS USING NEURAL NETWORKS

Ian Flood
School of Building and Estate Management

National University of Singapore
10 Kent Ridge Crescent
Singapore 0511

ABSTRACT

The paper describes the development of an efficient method of simulating construction activity using artificial neural networks. A brief introduction to construction simulation is provided, and a major limitation of the technique, that of unacceptably lengthy processing periods, is highlighted. This is followed by a description of the main mechanisms of neural networks and their potential as a means of increasing the execution speed of construction simulations. A modular neural network approach is proposed as the basis of a simulation facility suitable for general application to construction. The main types of modules required are identified, and the ways in which they can be implemented in neural circuitry are explored. The paper concludes with a general evaluation of the performance of the system, illustrating that it could operate many orders of magnitude faster than existing computer-based simulation facilities.

KEYWORDS: Neural Networks, Construction Simulation, Parallel Processing

1 INTRODUCTION

Computer-based simulation of construction processes is potentially a powerful modelling tool for use by managers/engineers in the planning, design, estimation, monitoring and control of all types of construction activity. Studies can be undertaken to determine, for example, the most appropriate construction method, plant combination and labour allocation, so as to optimize costs, time and production. In addition, simulation can be used to predict probable project durations, costs and production for estimating and tendering purposes, as well as to evaluate the likely influence on cost, time and production of unforeseen circumstances once work has commenced or been completed [1,2].

However, despite these merits, simulation is usually extravagant in its use of computer processing time. Even simple simulation models involve extensive computation and thus tend to involve lengthy execution. Consequently, it is often the case that simulation results cannot be produced quickly enough to be of practical benefit to managers, especially when the model being run is large and detailed. The problem is compounded by the fact that often many simulation runs need to be executed before a final answer can be established.

In order that the full benefits of simulation be realized, it is necessary to develop some method of processing simulation models that is many orders of magnitude faster than existing techniques. This paper presents an investigation of one possible way of achieving this aim through the use of artificial neural networks. Before describing the neural network approach, it is important to have a basic understanding of the key concepts concerning construction simulation modelling - a brief introduction is provided for this purpose.

2 CONSTRUCTION SIMULATION

Simulation is a dynamic process whereby a model (in this case a representation of a construction system) is subjected to a step-by-step change in its state, with respect to time. The state of the model following each step provides a description of the expected state of the real system at a corresponding point in time. The basis of any simulation study is the model of the system under investigation, the design of which usually starts with a schematic diagram/model. This is a graphic representation of the process flow and logic of a system (an excavation project in the case of Figure 1) providing the framework of the model to which more specific information (such as, the durations of activities and the capacities of storage facilities) will necessarily be added for the purpose of simulation. An understanding of schematic modelling is sufficient for readers of this paper.

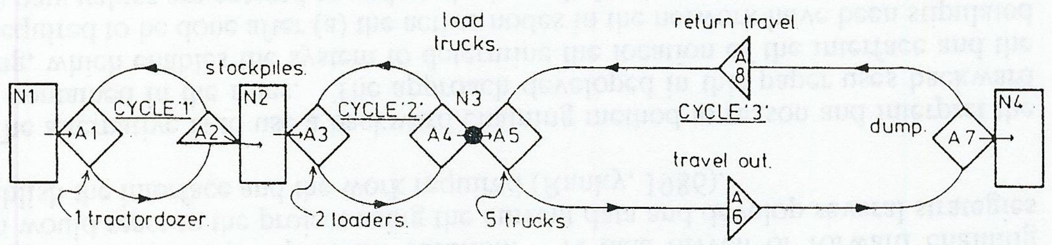


FIGURE 1: Schematic Model of an Excavation System

The ICONS [1,2,3] simulation system, as an example, provides five primary symbols for developing schematic models - these are shown in their simplest form in Figure 2. The first two symbols, the stationary and moving actqueues, each represent an operation, or a

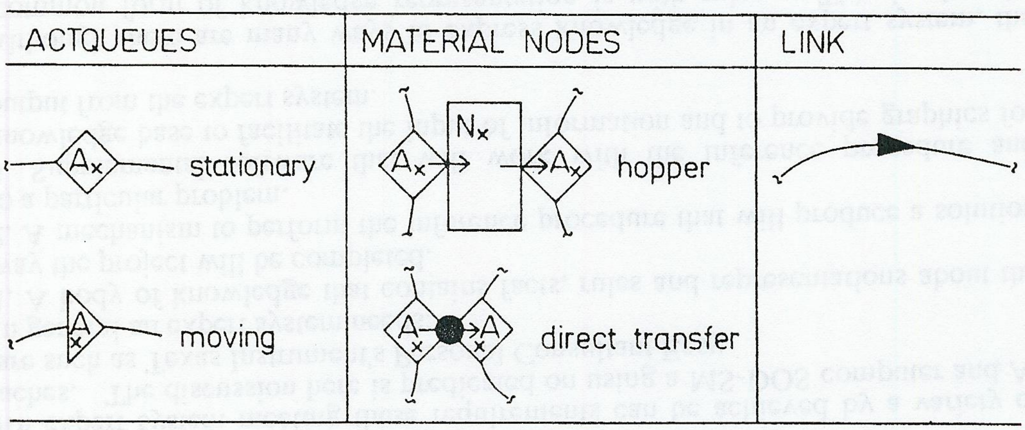


FIGURE 2: ICONS Schematic Modelling Symbols

discrete part of a process, performed by productive resources such as gangs of men, excavators or trucks. Actqueues have both an activity phase (representing the actual operation to be executed) which normally takes time to execute, and a queueing phase in which productive resources wait while they are prevented from starting the activity. Normally, actqueues are linked to form cycles of operations, as shown in Figure 1. The stationary actqueue is used to represent an operation of a stationary productive resource, such as a fixed batching plant, whilst the moving actqueue is for situations where the operation involves travel. The differentiation between the two symbols is solely for the benefit of the user in interpreting a schematic model and does not represent any difference in logic.

The third and fourth symbols shown in Figure 2, the material nodes, represent points in a system where material (such as, wet concrete, prefabricated units, or spoil) is transferred between productive resources on different activities. The hopper node has a facility for accumulating a buffer store of material between transferring productive resources. It can be used to represent, for example, a material store or an actual hopper, such as, a wet concrete hopper between a mixer and distribution trucks. The direct transfer node, on the other hand, represents a point in a system where material is transferred straight from one productive resource to another, with no provision for storage. An example would be where an excavator loads spoil directly into tipper-trucks. In both types of node, the arrow represents the direction of material transfer.

The final symbol shown in the Figure is the link, and is used to indicate the sequence in which actqueues are performed by productive resources. The schematic model shown in Figure 1 includes examples of all five symbols. The system comprises three cycles of actqueues, representing the respective processes of a tractor-dozer that pushes spoil into stockpiles, followed by two loaders that transfer the spoil to tipper trucks which in turn carry the spoil to a dumping point. Note that the material to be excavated is represented by a hopper node, N1, and acts as a point of material input to the system. Likewise, the point at which spoil is dumped is represented by a hopper node, N4, this time acting as a point of material output from the system. A hopper node is also used at the point of material transfer between the tractor-dozer and the loaders, N2, providing a temporary store for the spoil. The point at which spoil is loaded onto trucks is represented by a direct transfer node, N3, indicating that a loader cannot deposit its load until there is a truck available.

3 NEURAL NETWORKS

Typically, simulation models of the type described above are implemented as a serial algorithm on a general purpose digital computer, whereby each computing operation in a simulation run is executed in sequence by the computer. The serial nature of this method of computing means that the larger and more detailed a model the longer it takes to process. This problem is aggravated by the fact that simulation algorithms are in themselves relatively complicated and thus expensive in their use of processing time. An alternative method of computing that offers a solution to these problems is that of artificial neural networks. A neural network simulation system could, as will be expounded, operate many orders of magnitude faster than traditional computing implementations and with complete independence from the size or complexity of a model. The way in which such a system can be realized is described, along with a brief introduction to neural networks, in the following sections.

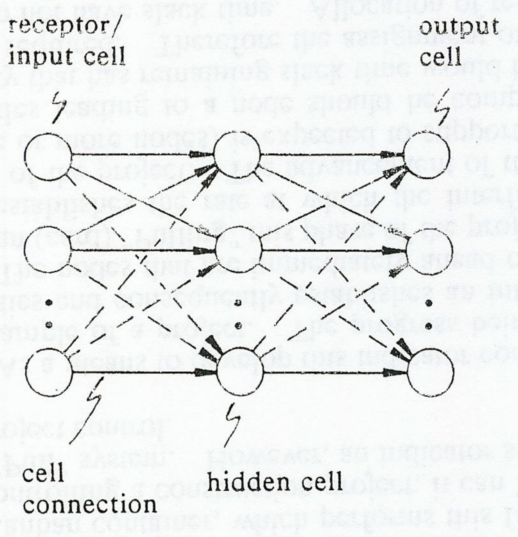


FIGURE 3: Simple Neural Network

Artificial neural networks are computing devices that model, to different degrees of exactness, the structure and operation of the central nervous system. They are configured from a large number of parallel operating neuron-like processing units, or cells, each of which performs some primitive function. These cells are inter-connected, as illustrated in Figure 3, forming a network that performs a collective higher-order function, such as the optimal sequencing of construction tasks [4]. Typically, information is input to a network through a set of receptor cells. This stimulates a response in the hidden cells, which in turn, stimulates the production of a result across the set of output cells.

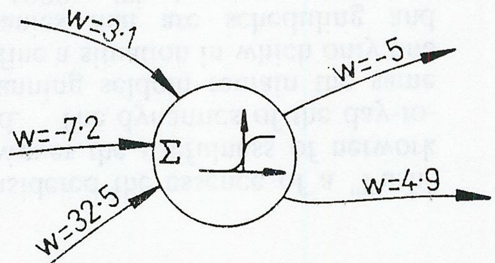
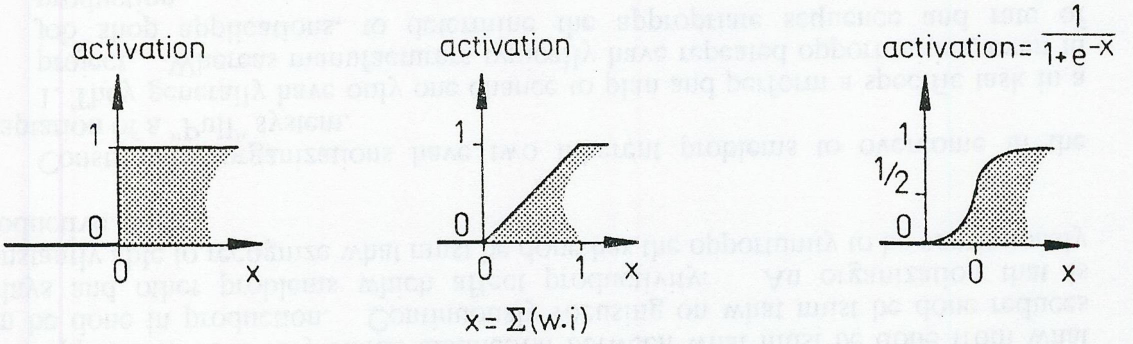


FIGURE 4: Single Neuron-Like Cell

The basic structure and mechanism of a cell can be understood by reference to Figure 4. At given points in time, a cell receives a value from each of its input connections which it then sums and puts through what is termed its activation function (examples of which are shown in Figure 5). This produces a level of activation for the cell which is then transmitted along its output connections to other cells in the network. Every connection, whether input or output, includes a weighting factor, *w*, which is used to multiply values transmitted across the connection. Essentially, it is the set of connection weights and the activation functions in a neural network that determine its overall function.



(a) Step Function (b) Sloping Step Function (c) Sigmoid Function

FIGURE 5: Examples of Commonly Used Activation Functions

The minimal amount of processing performed by each cell in a network coupled with the fact that all cells can operate simultaneously, are the reasons why neural networks are extremely fast information processing devices.

4 NEURAL NETWORK BASED CONSTRUCTION SIMULATION

One method of applying neural computing to construction simulation would be to assemble a single network for each simulation study from a number of network modules (or cell clusters) that have predefined circuits. Each of these modules would be designed to function as some primary component in a model, such as, a queueing facility, storage facility or productive resource. Once a model had been designed, the necessary modules would be linked (the output from a productive resource module would, for example, link-up to the input of a storage module) into a complete network description, and then downloaded to the neural computer.

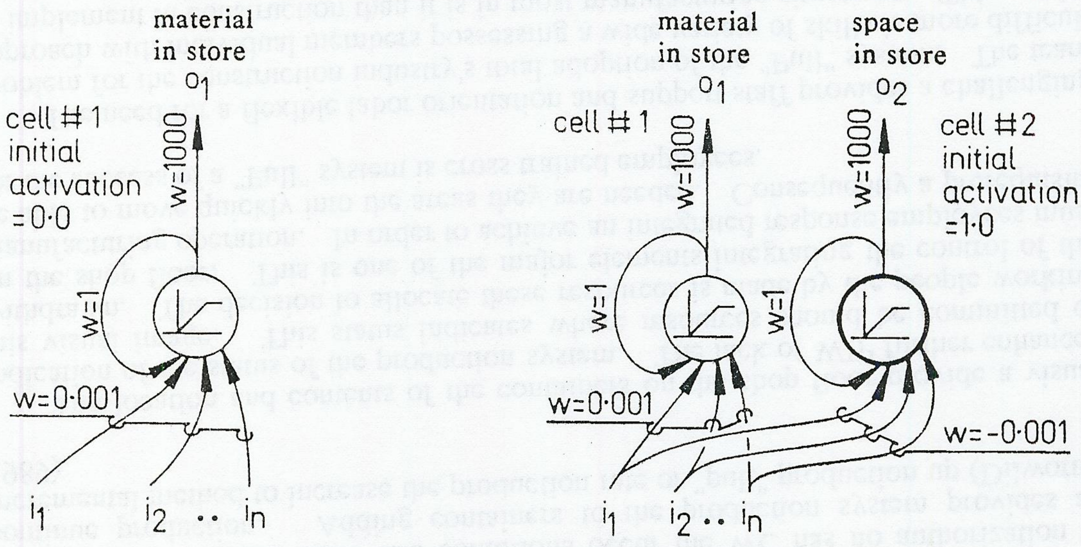
The completed network would be initialized by having all its inputs and levels of activation set to values corresponding to the initial state of the model. The network would respond to this by producing as output the state of the model corresponding to the next point in simulation time. This resultant information would, in turn, be transmitted back as input thereby triggering another response in the network and, subsequently, resulting in the output of the next state of the model. This process would be repeated until the state of the model at the final point in simulation time was reached.

Two distinct classes of module are required in construction simulation: these are, modules where the connection weights can be derived deterministically, and those where the weights have to be developed using some network training procedure. To illustrate these points, one example of each of these module types is described in the following sections. Although a complete simulation system may have as many as ten or twenty different types of network module, the two examples provided encompass all the relevant principles involved in circuit design.

4.1 Neural Circuits Representing Storage Facilities

The first module to be considered represents a storage facility for collecting material output from a system, such as, a spoil heap where excavated material is dumped by tipper-trucks (see component N4 in Figure 1 for example). It serves to illustrate the development of the type of module where connection weights can be derived deterministically.

The primary function of such a module is to keep account of the amount of material placed in the storage facility over time. This can be achieved simply using a neural network module of the form illustrated in Figure 6.a. At each simulation time step, the amount of material to be placed in the store is given by the input values i_1 through to i_n. In the spoil-



(a) material content (b) material content and remaining space

FIGURE 6: Neural Circuits Representing Storage Facilities for Material Collection

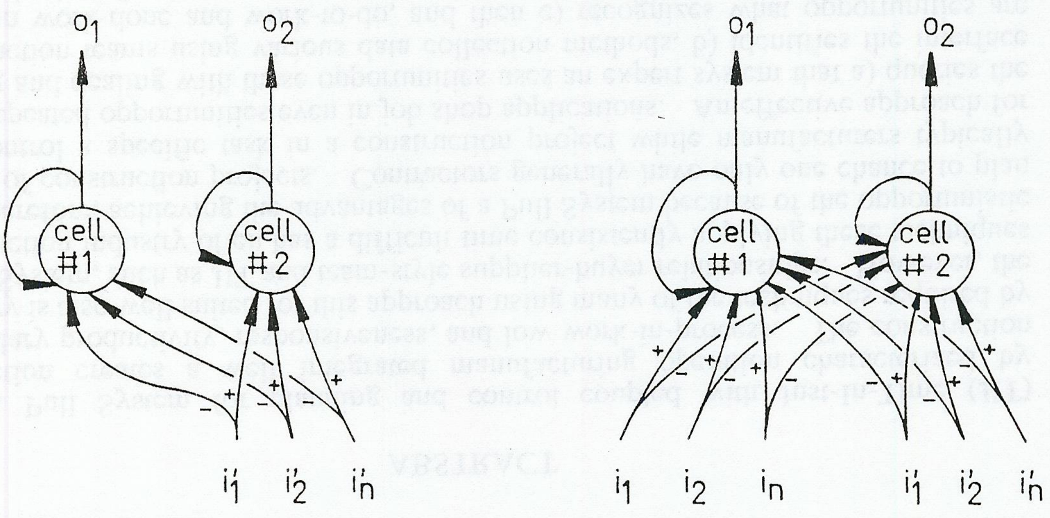
heap example, if there were five tipper-trucks in the system then there would be five input connections, each relaying the amount of material to be dumped by a different truck. The amount of material in the store at any point in time is given by the value output at o_1.

Inside the module, the amount of material contained in the store is measured as the cell's level of activation. This is limited to values between 0.0 and 1.0 since the cell uses the activation function shown in Figure 5.b. A value of 0.0 would indicate that the store was empty whilst a value of 1.0 would indicate that it was full. At each step in a simulation, the cells level of activation is fed back and added to all other inputs, the result of which is then placed through the cell's activation function. The new level of activation generated by this process represents the updated content of the store, taking into account the addition of new material at inputs i_1 to i_n.

The weights on the various connections have a number of functions. The weight on the output, o_1, modifies the measure of the current quantity of material contained in the store, so that it need not be limited to a value between 0.0 and 1.0. In this example, the value of the weight implies the spoil heap has a capacity range of 0 to 1000 cubic-metres. For the opposite reason, the weights on the input connections scale-down inputs, in this case to one thousandth of their original value. If necessary, the weights on the connections representing material input to the store could also be designed to adjust for expansion or compression of material during dumping.

The logic of the storage facility, as shown in Figure 6.a, has to be extended to furnish information concerning its remaining capacity. This information is used by productive resources (such as the tipper-trucks) to decide whether or not they should attempt to place material in the store. The addition of a second cell, as shown in bold in Figure 6.b, is required to achieve the extra logic. The second cell is similar to the first, but with a notable variation in its function. That is, the activation level of the cell provides a measure of the remaining capacity of the store and must, therefore, reduce as material is input. This is the converse of the mode of operation of the first cell, and is achieved simply by using negative weights on the connections representing the input of material to the store.

Situations where material is removed from, rather than input to, the store (such as occurs at node N1 in Figure 1) can be represented by the neural circuit shown in Figure 7.a. This is identical to the circuit in Figure 6.b, except that the weights on the input connections, i_1'



(a) supply storage facility (b) combined collection/supply storage facility

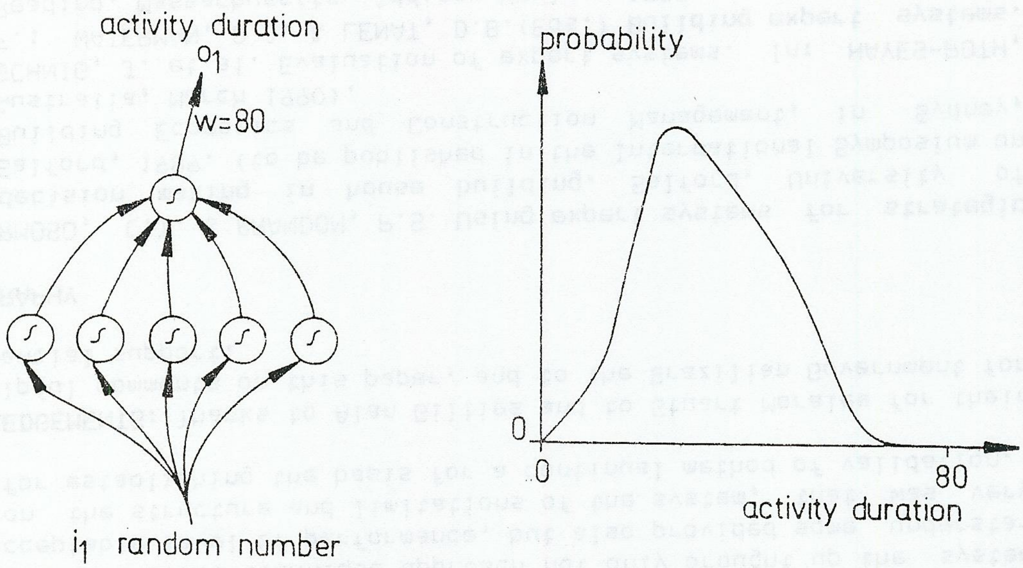
FIGURE 7: Representing Storage Facilities for Material Collection and Supply

to i_n', are of reverse sign. Thus, signals arriving along these connections would cause a reduction in the level of activation of cell #1 which represents the content of the store, and increase the activation of cell #2 which represents the remaining capacity of the store. Effectively, the input connections communicate information concerning the removal, as opposed to the addition, of material at the store, as required.

Figure 7.b shows a neural module representing a storage facility where material can be both added, and removed, such as occurs at node N2 in Figure 1. This is simply an aggregation of the circuits representing collection and supply type storage facilities shown in Figures 6.b and 7.a.

4.2 Neural Circuit for Generating Activity Durations

Typically, construction systems are fraught with uncertainty, making it difficult to predict performance and behaviour with any degree of accuracy. The effects of uncertainty as such can be taken into account by the use of stochastic/probabilistic simulation techniques, where it is recognized that the duration of any activity may vary each occasion it is repeated. Essentially, this involves the random selection of an activity duration from a representative distribution of values whenever the activity is due to commence. Implementing such a function in neural circuitry requires a fundamentally different approach to that described for the storage facilities above.



(a) neural circuit for generating durations (b) example probability distribution

FIGURE 8: Generating Activity Durations

Figure 8.a shows an example of a neural module that performs this type of function, in this case, producing activity durations that have a probability of occurrence shown by the distribution of Figure 8.b. Whenever the activity represented by this function is due to commence, a random number with a value between 0 and 1 is input at i_1. The network responds to this by outputting a value at o_1 which is read as the duration for the activity. Note that all cells use the sigmoid function shown in Figure 5.c

The problem is how to determine a set of connection weights that will make the module perform such a function. One way of achieving this is to employ an iterative training procedure. Each member in a set of inputs (in this case random numbers between 0 and 1) is presented in turn to the module, and the resultant output (representing the corresponding activity duration) is observed. The weights are then adjusted according to some rule so that future output will be closer to that required. This process is repeated many times until the module responds to all the input patterns in a satisfactory manner. For the module shown in Figure 8.a, weight changes were made in accordance with the Generalized Delta Rule [5], and required 2346 iterations to achieve output that was correct to within five percent of the training data.

A second problem is concerned with how many cells to utilize in such a module. For that shown in Figure 8.a, 5 cells was found to be sufficient, but as a general principle, the number of cells required, to achieve a given level of accuracy, increases with the complexity of the shape of the distribution curve being modelled. Beyond this, however, deciding the number of cells to use is essentially a process of trial-and-error.

As a final point, the module could be made more sophisticated, if necessary, to take into account the effects of factors such as weather and repetition on the time taken to perform an activity. This would, however, require a network far larger than that shown in Figure 8, and would likely prove extremely difficult, or even impossible, to train to a high degree of accuracy. One possible solution to this latter problem is a new type of network and complementary training algorithm [6] that provides rapid and guaranteed convergence on an appropriate set of connection weights. The technique has the added advantage of determining, as a matter of course, the number of cells required to achieve output that is within any chosen degree of accuracy.

5 DISCUSSION AND CONCLUSIONS

The paper has demonstrated the feasibility of implementing construction simulation models in neural circuitry. The primary benefit of the approach is fast processing, with no dependence on the size and detail of a model. This independence of processing speed from model complexity can be attributed to the parallel execution by the neural simulator of all concurrent simulation processes.

However, the absolute speed of execution of a simulation depends on the throughput or rate at which a neural network module can process information. Definite figures are not available for this as yet since neural hardware is very much in its early stages of development, though for the purpose of this discussion a conservative estimate of 10,000 to 20,000 input to output operations per second will be assumed. Since, in construction simulation, there is rarely any need to consider time periods shorter than a deciminute (six seconds) it should therefore be possible to simulate at least one day's work every second.

This amounts to a dramatic improvement over conventional computer simulation implementations, where a model the size of that shown in Figure 1 would take in the order of 5 minutes to simulate 24 hours work (using a PRIME 750 minicomputer [2,7] and assuming 14,400 time steps per simulated working day), while larger models incorporating one or two hundred productive resources would take several hours to simulate the same period. The profound improvement in performance offered by the neural network approach would enable the simulation of many more alternative construction methods and set-ups within a given period of time. This, in turn, would facilitate the selection of solutions to a construction problem that are highly optimal in terms of say time, cost, or production, and do so irrespective of the size and detail of the construction system under investigation.

REFERENCES

1. Flood I, *Construction Simulation Modelling using Serial and Parallel Processing Techniques*, PhD Thesis, University of Manchester, (1986).

2. Pilcher R and Flood I, "The Use of Simulation Models in Construction", Proceedings of the Institution of Civil Engineers, Part 1 Design and Construction, 76, pp 635-652, (August 1984).

3. Lewis J and Flood I, "Modelling and Simulation of Construction Operations using Interactive Computer Graphics", Proceedings of CIB W-65 Symposium, London, (September 1987).

4. Flood I, "A Neural Network Approach to the Sequencing of Construction Tasks", Proceedings of the 6th International Symposium on Automation and Robotics in Construction, San Francisco, pp 204-211, (June, 1989).

5. Rumelhart D E, et al, *Parallel Distributed Processing*, Volumes 1 and 2, (MIT Press, 1986).

6. Flood I, "A Surface Generation View of Neural Networks", Occasional Paper, School of Building and Estate Management, National University of Singapore, (1990), (in press).

7. Flood I and Pilcher R, "Increasing the Efficiency of Construction Simulation Modelling by using Parallel Processing", Proceedings of the 10th Triennial Congress of the International Council for Building Research, Studies and Documentation, 3, Washington D.C., (September 1986).